Logarithmic Space Reductions

We write

\[ A \leq_L B \]

if there is a reduction \( f \) of \( A \) to \( B \) that is computable by a deterministic Turing machine using \( O(\log n) \) workspace (with a read-only input tape and write-only output tape).

**Note:** We can compose \( \leq_L \) reductions. So,

if \( A \leq_L B \) and \( B \leq_L C \) then \( A \leq_L C \)
NP-complete Problems

Analysing carefully the reductions we constructed in our proofs of NP-completeness, we can see that \texttt{SAT} and the various other NP-complete problems are actually complete under $\leq_L$ reductions.

Thus, if $\texttt{SAT} \leq_L A$ for some problem $A$ in $L$ then not only $P = NP$ but also $L = NP$. 
P-complete Problems

It makes little sense to talk of complete problems for the class $P$ with respect to polynomial time reducibility $\leq_P$.

There are problems that are complete for $P$ with respect to logarithmic space reductions $\leq_L$.
One example is $CVP$—the circuit value problem.

That is, for every language $A$ in $P$,

$$A \leq_L CVP$$

- If $CVP \in L$ then $L = P$.
- If $CVP \in NL$ then $NL = P$. 

Anuj Dawar Complexity Theory
Circuits

A circuit is a directed graph $G = (V, E)$, with $V = \{1, \ldots, n\}$ together with a labeling: $l : V \to \{\text{true}, \text{false}, \land, \lor, \neg\}$, satisfying:

- If there is an edge $(i, j)$, then $i < j$;
- Every node in $V$ has \textit{indegree} at most 2.
- A node $v$ has
  - indegree 0 iff $l(v) \in \{\text{true}, \text{false}\}$;
  - indegree 1 iff $l(v) = \neg$;
  - indegree 2 iff $l(v) \in \{\lor, \land\}$

The value of the expression is given by the value at node $n$. 
A circuit is a more compact way of representing a Boolean expression.

*Identical subexpressions need not be repeated.*

CVP - the *circuit value problem* is, given a circuit, determine the value of the result node $n$.

CVP is solvable in polynomial time, by the algorithm which examines the nodes in increasing order, assigning a value `true` or `false` to each node.
Reachability

Similarly, it can be shown that Reachability is, in fact, NL-complete. For any language $A \in NL$, we have $A \leq_L \text{Reachability}$

$L = NL$ if, and only if, Reachability $\in L$

*Note:* it is known that the reachability problem for undirected graphs is in $L$. 

Anuj Dawar Complexity Theory
Our aim now is to show that there are languages (or, equivalently, decision problems) that we can prove are not in \( P \).

This is done by showing that, for every reasonable function \( f \), there is a language that is not in \( \text{TIME}(f) \).

The proof is based on the diagonal method, as in the proof of the undecidability of the halting problem.
Time Hierarchy Theorem

For any constructible function $f$, with $f(n) \geq n$, define the $f$-bounded halting language to be:

$$H_f = \{[M], x \mid M \text{ accepts } x \text{ in } f(|x|) \text{ steps}\}$$

where $[M]$ is a description of $M$ in some fixed encoding scheme. Then, we can show $H_f \in \text{TIME}(f(n)^2)$ and $H_f \not\in \text{TIME}(f(\lfloor n/2 \rfloor))$.

Time Hierarchy Theorem
For any constructible function $f(n) \geq n$, $\text{TIME}(f(n))$ is properly contained in $\text{TIME}(f(2n + 1)^2)$. 
Strong Hierarchy Theorems

For any constructible function \( f(n) \geq n \), \( \text{TIME}(f(n)) \) is properly contained in \( \text{TIME}(f(n)(\log f(n))) \).

**Space Hierarchy Theorem**
For any pair of constructible functions \( f \) and \( g \), with \( f = O(g) \) and \( g \neq O(f) \), there is a language in \( \text{SPACE}(g(n)) \) that is not in \( \text{SPACE}(f(n)) \).

Similar results can be established for nondeterministic time and space classes.
Consequences

• For each $k$, $\text{TIME}(n^k) \neq \text{P}$.

• $\text{P} \neq \text{EXP}$.

• $\text{L} \neq \text{PSPACE}$.

• Any language that is $\text{EXP}$-complete is not in $\text{P}$.

• There are no problems in $\text{P}$ that are complete under linear time reductions.